

Introduction

Recursive methods are typically very short, the real challenge is to come up with the solution in the first place. You are thus strongly advised to spend time solving such problems yourself. Of course, you must also be able to accurately trace recursive ~~and non-recursive~~ methods. Be sure to check the textbook (section 5.9 and chapter 13, in particular) and try the warm-up exercises on the Bits 'n Pieces page, before attempting the following exercises.

Exercises

- (a)** Write a recursive Java method that counts the number of occurrences of the character "e" in a string.
- (b)** Write a recursive method that takes a decimal value as an *int* parameter and prints out the binary equivalent of it on the console. For example, if your method is passed the integer value 6 (base ten) it should print 110.
- (c)** Suppose you have an ArrayList of String's. Write a recursive method that returns true if the strings in the array are in lexicographic (effectively alphabetic) order, and false otherwise.
- (d)** Write a recursive method that enumerates and prints all n-digit even numbers in which each digit of the number, from its most-significant to least-significant digits, is greater than the previous one (that is, for instance, the number 124 is ok, but the number 142 is not, since 2 is less than 4). Make sure your program prints each number just once. For example, if *n*=3 it should print 124 126 128 134 136 138 146 148 156 and so on up to 999. *Hint:* If you had a method that checked whether a number meets the stated criteria or not, you could use it to filter the candidate values you generated recursively, printing only those that meet the criteria ~ideally the criteria checking method would also be recursive.